note

Homework

Semaphore

S0 = 1, S1 = 1, S2 = 0
Process P0

```
while(true) {
    wait(S0);
    print '0';
    signal(S1);
    signal(S2);
}
```

Process P1

```
wait(S1);
signal(S0);
```

Process P2

```
wait(S2)
signal(S0)
```

TH1:

Process 0	Process 1	Process 2
wait(S0) -> S0 = 0	wait(S1) -> S1 = 0	wait(S2) -> S2 is 0 so it has to wait until S2 is 1
print '0'	signal(S0) -> S0 = 1	waiting
signal(S1) -> S1 = 1	done	waiting
signal(S2) -> S2 = 1	-	waiting
wait(S0) -> S0 = 0	-	signal(S0) -> S0 = 1
print '0'	-	done
signal(S1) -> S1 = 1	-	-
signal(S2) -> S2 = 1	-	-
wait(S0) -> S0 = 0	-	-
print '0'	-	-
signal(S1) -> S1 = 1	-	-
signal(S2) -> S2 = 1	-	-
waiting	-	-
- ...	-	-

TH2:

Process 0	Process 1	Process 2
wait(S0) -> S0 = 0	wait(S1) -> S1 = 0	wait(S2) -> S2 is 0 so it has to wait until S2 is 1
print '0'	signal(S0) -> S0 = 1	waiting
signal(S1) -> S1 = 1	done	waiting
signal(S2) -> S2 = 1	-	signal(S0) -> S0 = 1
wait(S0) -> S0 = 0	-	done

Process 0	Process 1	Process 2
print '0'	-	-
signal(S1) -> S1 = 1	-	-
signal(S2) -> S2 = 1	-	-
wait(S0) -> As S0 is 0 it has to wait	-	-
waiting	-	-
...	-	-
...	-	-
waiting indefinitely	-	-
- ...	-	-

Even though S1 and S2 is signaled many times, as they are binary semaphore, their values can as such not exceed 1.

So it prints 0 two or three times depending on the speed of the processes